

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification⁶ :

G06T 9/00, G06F 17/30

A1

(11) International Publication Number:

WO 95/34051

(43) International Publication Date:

14 December 1995 (14.12.95)

(21) International Application Number: PCT/US95/07210

(22) International Filing Date: 6 June 1995 (06.06.95)

(30) Priority Data:

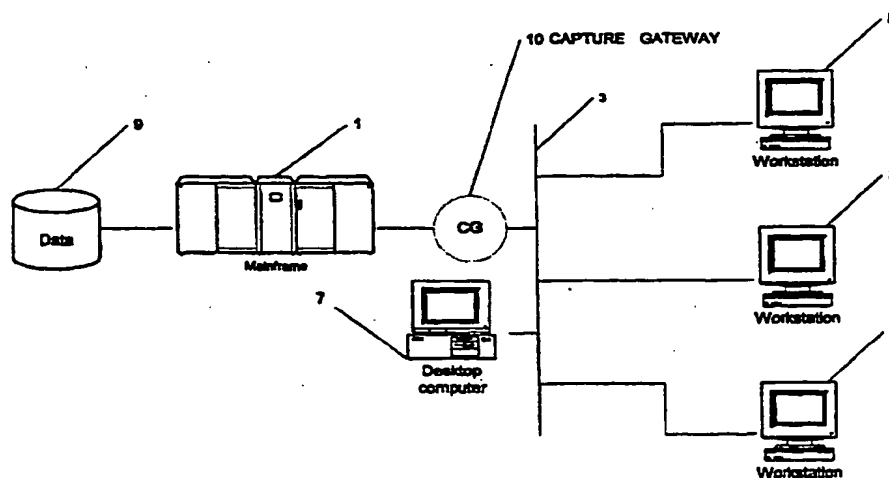
08/254,196

6 June 1994 (06.06.94)

US

(71) Applicant: SPECTRAGRAPHICS CORPORATION [US/US];
9707 Waples Street, San Diego, CA 92121 (US).(72) Inventor: EGHBALNIA, Hamid; W10203 Highway CS,
Poynette, WI 53955 (US).(74) Agents: LAND, John et al.; Fish & Richardson P.C., Suite
1400, 4225 Executive Square, La Jolla, CA 92037 (US).(81) Designated States: CA, JP, KR, European patent (AT, BE, CH,
DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).**Published***With international search report.**Before the expiration of the time limit for amending the
claims and to be republished in the event of the receipt of
amendments.*

(54) Title: METHOD AND APPARATUS FOR CAPTURING AND DISTRIBUTING GRAPHICAL DATA



(57) Abstract

A method and system for capturing graphic image in a universal format and distributing such information between computer system users for sharing graphics among systems having varying platforms and graphic image data types. A graphic image represented in an image data structure format is captured and converted into a universal graphic image data structure format that preserves available geometry and attribute information. The universal graphic image data structure format stores the geometry and attributes information essential to render captured graphic images essentially as their originating application omitting non-essential data. After creating or retrieving an arbitrary image, a capture mechanism converts the image data structure format to the universal graphic image data structure format (10). The universal format may be transmitted to a workstation (5) running the inventive system to render the universal format irrespective of the receiving user's graphic system and without the originating drawing application.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	GB	United Kingdom	MR	Mauritania
AU	Australia	GE	Georgia	MW	Malawi
BB	Barbados	GN	Guinea	NE	Niger
BE	Belgium	GR	Greece	NL	Netherlands
BF	Burkina Faso	HU	Hungary	NO	Norway
BG	Bulgaria	IE	Ireland	NZ	New Zealand
BJ	Benin	IT	Italy	PL	Poland
BR	Brazil	JP	Japan	PT	Portugal
BY	Belarus	KE	Kenya	RO	Romania
CA	Canada	KG	Kyrgyzstan	RU	Russian Federation
CF	Central African Republic	KP	Democratic People's Republic of Korea	SD	Sudan
CG	Congo	KR	Republic of Korea	SE	Sweden
CH	Switzerland	KZ	Kazakhstan	SI	Slovenia
CI	Côte d'Ivoire	LI	Liechtenstein	SK	Slovakia
CM	Cameroon	LK	Sri Lanka	SN	Senegal
CN	China	LU	Luxembourg	TD	Chad
CS	Czechoslovakia	LV	Latvia	TG	Togo
CZ	Czech Republic	MC	Monaco	TJ	Tajikistan
DE	Germany	MD	Republic of Moldova	TT	Trinidad and Tobago
DK	Denmark	MG	Madagascar	UA	Ukraine
ES	Spain	ML	Mali	US	United States of America
FI	Finland	MN	Mongolia	UZ	Uzbekistan
FR	France			VN	Viet Nam
GA	Gabon				

METHOD AND APPARATUS FOR CAPTURING AND DISTRIBUTING GRAPHICAL DATA

BACKGROUND OF THE INVENTION

1. *Field of the Invention*

5 This invention relates to computer graphics and video display systems, and more particularly to a method and system for capturing graphical information in a universal format and distributing such information between computer system users.

2. *Description of Related Art*

10 The advent of computers has fostered creation of a large number of interactive graphics drawing and illustration programs. Such programs typically define graphics images in one of three ways. So-called "paint" programs typically generate "bitmap" images that are stored as a 2-dimensional array of values representing picture elements, or "pixels", each having a particular color value. A second type of graphics program defines a graphics image as a data structure representing 2-dimensional images made up of lines, curves, planes, and similar structures positioned in the
15 same plane or in multiple parallel planes. A third type of graphics program defines a graphics image as a data structure representing a 3-dimensional model of one more objects. (It should be noted that 3-dimensional graphics data structures are ultimately rendered as 2-dimensional images in most cases, and that both 3-dimensional and 2-dimensional graphics images are generally represented on final output or display devices as bitmap images.)

20 With 3-dimensional systems, a complex object is modeled by breaking the object up into simple shapes (graphics primitives) that have associated attribute information. The information necessary to model a 3-dimensional object is typically kept in a "tree" type data structure in which the nodes of the tree define graphics primitives (such as polygons, vectors, surfaces, etc.), attributes (such as polygon color, line thickness, transformation matrices, etc.), and structural
25 information (such as pointers to related nodes). As is known in the art, by "traversing" the nodes of a tree that defines an object, a 2-dimensional representation of the 3-dimensional model can be rendered and output for display. An example of a hardware-oriented graphics display

-2-

processor is set forth in U.S. Patent No. 4,862,392 entitled "Geometry Processor for Graphics Display System". Unlike 2-dimensional image data, 3-dimensional image data can be used to view a modeled object from essentially any angle by rendering the model from different angles. Thus, while 2-dimensional objects can only be translated across a viewing area, or rotated in the image plane (i.e., around the "Z" axis), 3-dimensional models can be rotated around any of the X, Y, and Z axes.

Until the late 1960's, interactive graphics required expensive workstations and host processors. Such systems were typically available only to a few sophisticated users. The advent of relatively inexpensive time-sharing systems with video terminals had a dramatic impact on the use of interactive graphics drawing systems. Applications involving interactive graphics were now available for a large number of users, and new and different techniques were developed for working with graphical images. As the cost associated with graphics systems decreased, the number of different graphics drawing programs increased. At present, a relatively large number of graphics drawing programs are in use. Each of these programs typically uses its own approach and technique for creating, processing, and representing image data, whether for display or for storage. For example, the AUTOCAD system developed by Autodesk, Inc. allows users to draw such items as mechanical and electrical components and systems. The images created by the AUTOCAD system are stored as DXF data files having a format particular to the AUTOCAD system. The DXF data format is designed to record information particular to the images created, including, for example, the geometric information necessary to model a 3-dimensional object such that the object can be rendered as a 2-dimensional image on a particular display device. However, many other graphics drawing programs use other graphics data structure formats.

A problem that arises with the existence of disparate graphics drawing programs and proprietary graphics data structures is in sharing graphics images among users. Although image storage, processing, and display works relatively well in a self-contained graphics system, problems arise when users attempt to share graphics images across systems. Prior art graphics programs suffer the disadvantage of being generally unable to share images among different systems having varying data types. Because data types are typically stored in data structure formats particular to the application that created the images, graphics images generally can only be directly rendered on platforms having that application program.

-3-

Another problem with a lack of a common graphics data file format is the lack of compatibility between different graphics drawing systems. Currently, there are a wide variety of devices and systems on the market for input and output of graphical image data. Depending on whether an application environment is simple or complex, an operator may be working on a single input/output (I/O) device, or using a number of devices. For example, input devices may include a digitizer tablet, a keyboard, a mouse, a light pen, or similar pointing and input devices. Output devices may include dot matrix printers, raster scan printers such as laser and ink jet printers, video monitors, and plotters. The aim of a standard computer graphics image representation in all of these environments should be to allow programs to make maximum use of both the total environment available to users and the specific characteristics unique to each device.

As a result of this lack of data structure commonality, to view a graphics image, a user generally must run the same application software which was used to create and store the graphics images. Yet the user may not be skilled in operating the original application program. Further, due to usage limitations for some network software or equipment incompatibilities, the user may not even be able to run the original application program from the user's computer station. Consequently, very little sharing of images is possible between different computer systems running different application software. Therefore, a need exists for an image capture and distribution system capable of sharing graphics images between different graphics systems supporting user's with varied skills.

In the past, attempts have been made to solve this problem by standardizing computer graphics and image data structures. For example, in 1982 an international group of graphics experts proposed a draft international standard to the ISO, referred to as the Graphical Kernel System (GKS). GKS is a graphics system which allows programs to support a wide variety of graphics devices. It is defined independently of programming languages and application programs. Unfortunately, the GKS system has not gained wide acceptance in the industry due to its inherent inability to accurately represent all possible geometries.

Lacking a common data structure format, in the past three different techniques have been used for sharing graphics image information among different users. One type of system is designed to translate stored graphics image data files from one format to another. For example, it is well

known to use translation programs to convert data stored in the AUTOCAD DXF format into data for use in the CADAM format, and vice versa. However, the number of translation programs required for essentially universal access to proprietary data files is great, due to the number of potential target data types (approximately 100 at present) available. Indeed, for universal translation directly from type to type, the number of translating programs required grows in a non-linear fashion. For example, to directly translate between 10 different formats requires 45 translation "filters". To directly translate between 50 different formats requires 1,225 filters.

Another approach to translating graphics data file structures is to use a "hub and spokes" approach of indirect translation. Using this technique, the data structure of a particular application program is passed through a first filter and converted to a common format. Next, the converted data in the common data format is passed through a second filter and converted to the desired target format. A graphics data image can then be rendered on a display device at the requesting platform. However, in present systems much of the original geometric information within the original graphics data structure is lost during the double filtering process. As a result, the final translation is typically of poor quality with limited ability to further manipulate the captured image. Moreover, to directly translate between 50 different formats requires 50 filters.

A drawback common to all filtering mechanisms is the need of filtering mechanisms to make a "guess" as to the actual functions performed by the originating application in rendering geometric data files. For example, an application may insert a series of points in a data file to indicate the shape of a spline. However, for the correct shape of the spline to be rendered, additional information coded in the application program may be necessary. This additional information may include: the type of spline, the generating functions used, the order of the spline, the end-point conditions, etc. Without some or all of this information, the filter program needs to "guess" at the nature of the data. Some guesses may be wrong, thereby producing less useful or useless data.

A totally different approach from translating source graphics data structures is to capture the bitmap output of the display device, regardless of the original data structure that created that bitmap, and communicate the captured bitmap to another user. This is essentially a "snapshot" approach, which produces a static image. The principal disadvantages of this approach are that all geometry information is lost, and only limited manipulation of the bitmap image can be

-5-

performed by the recipient. For example, if the original data structure represented a 3-dimensional model, the user receiving a captured bitmap from that model is no longer able to perform out-of-plane rotations or manipulations of the underlying geometry for the modeled object.

5 Therefore, a need exists for a universal graphics image capture and distribution system which allows users working with graphics images having varying "native" data structures to communicate with each other and share graphics image data. It would also be desirable that such a system provide an efficient and economic method of capturing and distributing such universal graphics images. The present invention provides such a graphics image capture and distribution
10 system.

SUMMARY OF THE INVENTION

The present invention solves the problems encountered in prior graphics imaging programs by providing a graphics image capture and distribution system that enables systems having varying platforms and graphics image data types to share images regardless of which system created the images.

Many prior art 2-dimensional and 3-dimensional graphics drawing programs convert stored graphics image data files into an "intermediate" data structure format prior to display. Although there are approximately 100 different stored graphics image data types, many prior art graphics imaging programs use only one of a handful of intermediate data structure formats, such as the well-known IBM 5080, PEX, and OpenGL formats. The present invention takes advantage of this fact by capturing a graphic image represented in an intermediate data structure format and converting the captured information into a universal graphic image data structure format that preserves as much geometry and attribute information as is available. The present invention requires only one translator for each of the small number of intermediate data structure formats presently in use.

This approach solves the additional problem of "guessing" faced by prior art filter programs. The application program supplies all information needed to render a graphics data file to an intermediate data structure format. Accordingly, that information is available to accurately convert the intermediate data structure format to the universal graphic image data structure format used by the present invention.

The universal graphic image data structure format used by the present invention stores the essential geometry and attribute information required to enable captured 2-dimensional and 3-dimensional graphics images to be rendered essentially as they were rendered by the originating graphics drawing program. Preferably, non-essential data that is not required for rendering purposes is stripped away. However, the universal graphic image data structure format provided by the present invention is sufficiently rich to represent all possible image geometries, yet is sufficiently simple and compact to reduce processing and storage requirements.

5 The present invention comprises a system for capturing graphics images and distributing those images between graphics systems located on a network (local or wide area), or between which data can otherwise be communicated (e.g., via modem or media transfer). The inventive system is preferably implemented in software which is executed on a processor having access to the graphics systems. The system uses an architecture which allows various graphics systems to communicate with each other and share image data over the network. The architecture provided by the present invention imposes constraints upon the creation, processing, transmission and storage of image data. These constraints ensure that any image can be rendered on any device on the network operating under the inventive system, regardless of the type of display device and regardless of the type of image or the originating graphics drawing program.

10 Using the present graphics image capture and distribution system, a user may capture images on the user's display device for transmission to another user on the network. To initiate an image capture, the user first creates or retrieves an image using a conventional graphics drawing program. For example, a user may create an image of a mechanical part using the CADAM drawing program. While this program uses a proprietary data format for storing graphics images, it represents such images on some computing platforms in the IBM 5080 intermediate data format. The user then initiates the capture mechanism of the present invention, which converts the intermediate data format of the image to a universal 3-dimensional image format. The universal format may then be transmitted to the receiving or "end" user. The end user may then "view" and manipulate the 3-dimensional image on a display device running under the inventive system by rendering the graphics image represented in the universal format to a display format compatible with the receiving graphics system. The end user does not need to know or have the CADAM application to see the captured image.

25 The present invention also has a "snooping" mechanism which records information about a graphics image as it is created using the originating graphics drawing program. This recorded information is discarded if the created image is never captured by the present invention. However, if an image is captured as described above, the "snooped" data is included with the captured graphics image data to provide initialization and other information necessary to properly display the captured image.

-8-

Because geometry and attribute information is preserved for most graphics images captured by the present invention, various rendering and transformation algorithms can be used to process graphics images captured and converted to the universal format. For example, images can be transformed using merging, warping, color change, morphing, spot-lighting, blending, colorizing, planar and non-planar rotation, zooming, and panning algorithms. Accordingly, the recipient of a graphics image captured by the present invention can manipulate the image without having access to or familiarity with the originating drawing application.

The details of the preferred embodiment of the present invention are set forth in the accompanying drawings and the description below. Once the details of the invention are known, numerous additional innovations and changes will become obvious to one skilled in the art.

BRIEF DESCRIPTION OF THE DRAWINGS

FIGURE 1 is a block diagram depicting the architecture of a prior art computer network system of the type commonly used for creating graphics images.

5 FIGURE 2 is a block diagram of the architecture shown in FIGURE 1, as modified to include a capture gateway function in accordance with the present invention.

FIGURE 3 is a pictorial representation of a monitor screen showing an example of usage of the present invention.

FIGURE 4 is a diagram of the components of the capture gateway function of the present invention.

10 FIGURE 5 is a flowchart of the basic steps for capturing a graphics image using the present invention.

FIGURE 6 is a diagram of the preferred method of representing a surface of revolution in accordance with the present invention.

FIGURE 7 is a diagram of an un-optimized data structure typical of the prior art.

15 FIGURE 8 is a diagram of an optimized data structure in accordance with the present invention.

FIGURES 9, 10, and 11 are pictorial representations of a monitor screen showing in sequence an example of usage of the hyper-connection feature of the present invention.

Like reference numbers and designations in the various drawings refer to like elements.

DETAILED DESCRIPTION OF THE INVENTION

Throughout this description, the preferred embodiment and examples shown should be considered as exemplars, rather than limitations on the present invention.

Overview of Environment

5 FIGURE 1 is a diagrammatic representation of a prior art network system of the type commonly used for creating graphical images. A central processor 1 (which may be, for example, a mainframe computer, a minicomputer, or a file server computer) is connected over a network 3 to a variety of workstations 5 and/or personal computers 7 (generally, computer stations). The central processor 1 is commonly coupled to a large data storage system 9, which is frequently
10 a repository for shared data files. In a typical system, a graphics application program would be resident on the central computer 1, but be accessible to users at the computer stations 5, 7. However, as is known in the art, application programs may be distributed essentially anywhere within the network system, and be accessed by other users on the system having the requisite privileges. Thus, an application program may be resident on local storage at one workstation 5
15 and accessible to users at other workstations 5 or the desktop computer 7.

A problem in such a system is that a user on one of the computer stations 5,7 may create a graphical image which he or she wishes to share with a different user. However, under most present systems, a second user on another computer station 5, 7 cannot display images created by the first user unless the second user has the knowledge and the capability required to run the
20 same application program that was used to create the graphics image to be shared.

FIGURE 2 is a diagrammatic representation of a conceptual method of implementing the present invention. FIGURE 2 is essentially similar to FIGURE 1, but with the addition of a "capture gateway" function 10 in the system. The capture gateway function 10 is preferably implemented as a computer software program, configured to run either on a single processor, or on distributed
25 processors. The computer program is preferably stored or storable on a storage media or device readable by a computer, for configuring and operating the computer when the storage media or device is read by the computer, the computer being operated to implement the capture and transformation functions of the present invention.

-11-

Principals of Operation

5 The present invention takes advantage of the fact that many graphics application programs generate a standard intermediate form of data structure from their native data storage files. Examples of such intermediate data structures are those that comply with the IBM 5080 graphics protocol, the OpenGL protocol defined by Sun Graphics, Inc., and the PEX protocol defined by the X-Consortium for use with X-Windows-based computer systems.

10 The present invention also takes advantage of the realization that such intermediate form data structures represent a transformation from an original data storage data structure. Thus, the developer of each graphics drawing program has already created a high-quality translation program that generates an intermediate form data structure compatible with supported display devices. The present invention therefore captures geometry and attribute information from an intermediate form data structure and translates that structure to a universal data structure format. A capture is defined as a computer session during which application-generated information is collected to the extent that such information can then be used, directly or indirectly, to replicate the presentation of that information. Such information may be obtained by (1) "wiretapping" the output of the graphics drawing application, (2) querying the application or application related data structures, (3) querying the device to which the application is connected, or (4) any appropriate combination of the above.

20 To facilitate such captures, the intermediate data structures useful to the present invention should have at least the following characteristics:

- (1) the data is stored in a file or is rendered through a wire protocol;
- (2) the data comprises distinct 2D or 3D geometric information;
- (3) the geometric information is accompanied by attributes for proper rendering;
- (4) the data content is sufficient to provide a consistent data set, and is reasonably application invariant.

25 The preferred embodiment of the present invention also provides a variety of device interfaces that allow geometry and attribute data in the universal format to be displayed on output devices that may not support the originating drawing program.

-12-

In operation, the capture gateway function 10 is activated by a user to (1) capture geometry and attribute data from one such intermediate form of graphics image representation and (2) translate the intermediate form to a universal graphics data structure. The information retained in the universal graphics data structure may be a mix of original application information and derived or transformed information. This universal data structure can then be transmitted by a first user to a second user. The second user is then able to display and manipulate the graphics image captured by the first user without having knowledge of or access to the original application program used to create the original drawing. While some of the functionality of the original graphics program may not be available (e.g., parts list generation), as much graphics information as is needed to faithfully reproduce a manipulable image is captured. Further, because the invention works on the intermediate data files produced by many graphics drawing programs, images from several such graphics programs may be captured and stored as a "compound document" and transmitted to a second user, who can effectively manipulate all of the captured images independently of the original application programs used to create them. Moreover, because only a few intermediate form data structures are in common usage, the invention can be implemented relatively easily.

Image Capture User Interface

FIGURE 3 helps explain the basic principles of operation of the capture gateway system of the present invention. FIGURE 3 presents a view of a terminal screen showing a graphical user interface as a user creating a graphics image might see it. A drawing application window 30 is shown within a larger workstation screen 32. In the example shown, the user has created an image of a satellite dish using a conventional graphics drawing program such as CATIA by Dassault of France. This particular program creates an intermediate data structure that conforms to the IBM 5080 protocol. Assuming that the user wishes to capture the graphics image shown in the drawing application window 30, the user uses a form of "drag and drop" interaction to drag an icon 34 representing the capture gateway function. (The capture gateway icon 34 is shown as a clipboard; other symbols could be used as desired). The capture gateway function would previously have been loaded some time during the user's session, either upon start-up or by the user manually selecting and running a program implementing the capture gateway function. Normally, however, a capture gateway icon 34 would be available to the user as a "tool". (Of course, means other than the use of an icon 34 to invoke the capture gateway functionality can

be used, such as drop-down menus, function key commands, pop-up menus, or command line entries.) In the example shown, the user would drag the capture gateway icon 34 onto the drawing application window 30 and "drop" it to invoke the capture gateway function 10. As one option, the entire contents of the drawing application window 30 would be captured. As another
5 option, the user selects a square or rectangular portion of the graphic image within the drawing application window 30, and thus generates a subset of the displayed graphics image as the subject matter of the capture. In an alternative embodiment, the graphics image could be dragged and dropped onto the icon 34 to invoke the capture gateway function 10.

Once a capture has been completed, the captured image may be shown in a second window (not
10 shown), or an indication given to the user that the capture has been completed. Thereafter, the user may transmit the contents of the capture window to another user (for example, over a network). The recipient user would have to have the rendering portion of the capture gateway function 10 running on his or her system to view and manipulate the received data structure. However, the recipient would not have to know the details of the program used by the original
15 user to create the drawing. Thus, by learning to use the capture gateway function 10, a user can view and manipulate drawings generated by others using a variety of different graphics drawing programs. The recipient user would then be free to annotate the received captured graphics image, transmit it to a suitable output device, or add other captured graphic images to the received data structure for retransmission to another user. Accordingly, a number of different
20 users throughout an enterprise may view and annotate or change images from disparate graphics drawing programs in a highly efficient and economical manner.

Since the universal data format used by the present invention captures all the geometry and attribute information available from the underlying application that created an image, the captured graphics image can be manipulated using that geometry and attribute information. Thus,
25 the present invention provides a far superior image than the mere capture of a bitmap. Because the number of intermediate graphics data structures that need to be converted to the universal data structure of the present invention is limited, the present invention provides an economical way of translating the graphics image data to a common format and making that format available to multiple users.

Preferred Implementation

The general architecture of the preferred embodiment of the present invention is shown in FIGURE 4. A workstation 5 is shown coupled to an interaction component 40 which provides the user interface functions described above. The interface component 40 is coupled to a persistence component 42 that is responsible for storing captured images as objects and providing transparent access to such objects within the system.

The interaction component 40 and persistence component 42 are coupled to an application component 44 which provides domain specific knowledge for the system. For example, the application component 44 may contain information particular to a specific display device and/or graphics image system. Thus, if the application component 44 determines that the user has an X-Windows display device, then the captured graphics data is rendered as an X-Windows compatible bitmap and transmitted to the user's monitor for display. The characteristics of a user's computer station can be determined in known manner by a query over the network, or by preset definition in a table referenced by network address.

The inventive system also includes a capture gateway (CG) intercept function 46 which monitors data communications between a graphics application program and a workstation 5 over the network 3 such that it can "snoop" or capture initial rendering-related information that such a graphics application program may generate and/or transmit to the workstation 5 at the beginning of a session. More particularly, the CG intercept function 46 obtains graphics data from the drawing application such that the graphic image being displayed at the time of capture can be regenerated from the captured data. Such information may vary among different intermediate data structures. With respect to the IBM 5080 protocol, at least the following resources comprise the graphics data set useful or necessary to a capture in the preferred embodiment of the present invention:

- (1) display list memory page;
- (2) attribute register sets, buffer address registers, and stack registers;
- (3) memory area control table;
- (4) regeneration address;
- (5) view port, clipping bounds, perspective depth, current transformation matrix;
- (6) color table;

-15-

- (7) 5080 line patterns, blink patterns, and area fill patterns;
- (8) programmable character set;
- (9) graphics interface registers.

5 This information is obtained in known fashion by querying the appropriate registers and tables defined under the 5080 protocol as implemented by particular vendors.

The CG intercept 46 also functions to actually capture the data stream necessary to obtain the intermediate form data structure of a designated graphics image for transformation to a universal data structure format.

10 The interaction component 40 is also preferably coupled to a capture gateway (CG) locator function 47, which serves to find an appropriate capture gateway function 10 in the network system when a capture is initiated by a user. This permits the capture gateway function 10 to be available to multiple users throughout a network system, without requiring multiple copies of the capture gateway function program modules for actually carrying out a graphics image capture. Accordingly, one capture gateway function 10 can be provided for multiple users, who invoke
15 shared components of the capture gateway functionality by use of the CG locator 47. Further, one capture gateway function 10 can accommodate captures from multiple graphics applications as long as they use the same intermediate form data structure.

Coupled to the CG intercept function 46, the CG locator function 47, and at least one persistence component 42 is a CG processing function 48, which actually performs the transformation from
20 an intermediate form data structure to the universal data structure format used in the preferred embodiment of the present invention.

Although the preferred embodiment of the present invention separates the CG intercept function 46 from the CG processing function 48, in an alternative embodiment, the two functions may be combined, and each computer station 5, 7 can directly access a "super" CG processing/intercept
25 function that can capture and transform a plurality of different intermediate data structures. However, the split of function described for the preferred embodiment has the advantage of ease of maintainability and economy of implementation.

Further, the persistence component 42 may be coupled to the CG processing function 48 and/or to the application component 44 by virtual connections 45. A virtual connection is similar to a device driver, in that it provides a standard interface between two programs and/or devices. In the preferred embodiment, providing virtual connections 45 as shown simplifies communications with the persistence component 42 and allows changes to be made to various components and functions without changing coupled components and functions. A further advantage of a virtual connection 45 is that it permits the persistence component 42 and the CG processing function 48 to be located on separate platforms.

FIGURE 5 is a flow chart of the basic steps for capturing a graphics image using the present invention. Assuming that the capture gateway function 10 is available to a user of a computer station 5, 7 as shown in FIGURE 2, a user may issue a capture request using the "drag and drop" graphical user interface described above (STEP 500). The use of "drag and drop" interfaces is well known in the prior art, and this action by the user results in invoking the capture gateway function 10. The user interface may also permit the user to designate a subportion of a displayed graphics image to be selected, in known fashion (e.g., by using a "selection box" to designate opposing corners of a rectangle).

After a user commences a capture, the capture gateway function determines the capture type (STEP 502). In the preferred embodiment of the present invention, this step is performed as follows:

- (1) Determine the window identification for the drawing application window 30 selected by the user by querying the operating system.
- (2) Normally, the returned window identification includes the name of the application and intermediate geometric data structure type running within the window. If so, proceed to STEP 504. If not, query the operating system (usually via a process table maintained by the operating system) to determine the application running in the window.
- (3) Based upon the identity of the application running in the drawing application window 30, determine the intermediate geometric data structure type used by that application. This is generally done by a simple look-up table, which maps known drawing application

-17-

programs to known intermediate data structure types (e.g., the CATIA program is known to use the IBM 5080 format).

Once the system determines the intermediate data structure form to be captured, the next step is to locate an appropriate capture gateway (STEP 504). This is done as follows:

- 5 (1) A query is sent over the network 3 via the CG locator 47 for a CG processing function 48 that can process the data structure type determined in STEP 502. A query also includes information regarding the user's equipment configuration (e.g., display type, display processing capabilities, etc.). This latter information is typically retained in a table which is configured upon installation (and changed periodically as needed) to map
10 each computer station 5, 7 with its equipment types. The query is actually in two parts: (a) can the receiving CG processing function 48 handle this intermediate data structure type, and (b) is the receiving CG processing function 48 available. Generation and transmittal of such queries are well known in the art. This division of function permits a plurality of CG processing functions 48 to exist on the network system, each with the
15 ability to translate one or more data types. Such an architecture also is particularly suited to a distributed processing environment.
- (2) If no CG processing function 48 is available, a dialogue message is displayed to the user to display various options, such as "retry" and/or "quit".
- 20 (3) If a CG processing function 48 is located that is available and can handle the data structure type, processing continues at the next step.

The next step is to "bind" the located capture gateway (STEP 506). In this process, the CG processing function 48 located in the previous step sends an available signal to the interaction component 40 handling the capture request issued by the user. The interaction component 40 queries the CG processing function 48 for detail information, such as the conversion capabilities
25 and version compatibility of the CG processing function 48. The interaction component 40 determines if the CG processing function 48 is satisfactory (e.g., capable of translating the data structure format of the application which is to be captured). If so, the interaction component 40

-18-

sends a signal to the CG processing function 48 that "binds" that function, thereby temporarily locking out all other users from accessing that function.

The next step is to actually initiate the capture (STEP 508). This is done in the preferred embodiment as follows:

- 5 (1) Any necessary initial information regarding the graphic drawing application from the "snooped" data obtained by the CG intercept function 46 is sent to the CG processing function 48.
- 10 (2) If the application from which the graphics image to be captured is running in the "immediate mode" (meaning that graphics data is generated "on the fly" for display on the user's monitor), the CG intercept function 46 generates a "REDRAW" command to the underlying graphics application. This causes the underlying application to regenerate and retransfer the graphics information necessary to redraw the selected image on the user's screen. However, the retransmitted data stream is instead captured by the CG intercept function 46. The returned data is transferred to the CG processing function 48 as a display list.
- 15 (3) On the other hand, if the underlying graphics application is running in "structure store mode" (meaning that the data used for generating a display image is stored in a temporary data structure accessible by the graphics application), the CG intercept function 46 generates a query that mimics a query from the underlying graphics application, and thereby accesses the temporary data structure directly. The returned data is transferred to the CG processing function 48 as a display list.
- 20 (4) Thereafter, the interaction component 40 either opens or uses an existing channel to the persistent component 42 in which the captured data will be saved after transformation.
- 25 (5) The interaction component 40 transmits the address of a "container" in the persistent component 42 to the CG processing function 48. A container is maintained by the persistence components 42, and is simply a form of temporary storage that is readily accessible by the capture gateway function 10. The CG processing function 48 uses the

container to deposit objects resulting from the conversion from the intermediate data structure to the universal data structure.

5 The next step is to process and deposit the captured image into the container (STEP 510). The data captured during the previous step is stored in known fashion in a tree or graph type data structure or display list, in which the nodes or entries define graphics primitives, attributes, and structural information as objects, in accordance with the following steps:

- 10 (1) The captured display list data is traversed, again in known fashion, with each entry being examined to see whether it can be transformed into a comparable data type within the universal data structure. Some entries may not relate to geometry or attributes at all, but to other information which is unnecessary to rendering a depiction of the modeled object as a graphics image. Other entries may describe proprietary types of data structures that are not supported in the universal data format, and thus cannot be transformed. In either case, these proprietary entries, and any referenced "children" entries that do not depend from other entries that can be transformed, are not included in the output data structure and are not processed further.
- 15

In an alternative embodiment, the CG intercept function 46 examines the captured data stream for nodes that are necessary for the display of graphics data, and omits transference to the CG processing function 48 of any nodes that do not affect the display of graphics data.

- 20 (2) As each node or entry that can be processed is traversed, translation is made from the geometry or attribute data of the captured image to comparable geometry or attribute data in the universal format (one example of such a transformation is set forth below). Thus, for example, if a node represents a line, a comparable representation of a line is stored in a similar node in the universal data format. The entire transformed data structure is stored in the container selected in STEP 508. The output from the CG processing function 48 is a contiguous display list along with a set of necessary resources. Preferably, the display list contains only graphics primitives, attributes, and viewing information.
- 25

-20-

- (3) As desired, an intermittent progress report may be sent back to the interaction component 40, which causes an appropriate message to be displayed to the user.
- (4) When transformation of the captured graphics data in its intermediate format to the universal data structure format is complete, a "DONE" message is sent to the interaction component 40 for display to the user.
- (5) The interaction component 40 then releases the CG processing function 48 that had been "bound" during STEP 506.

In effect, the CG processing function 48 that is capable of processing the captured intermediate data structure effectively emulates an interpreter capable of traversing the intermediate data structure and processing the display list represented by such a structure. The emulation interprets the captured display list, but without sending any primitives down a graphics pipeline. Instead, each entry of the display list is translated to a new, universal data structure representation.

Transformation of Data

In the preferred embodiment of the present invention, the geometry data of a captured image falls into one of six categories, as follows:

- (1) zero parameter entities representing points;
- (2) single parameter entities representing lines and curves;
- (3) two parameter entities representing planes and surfaces;
- (4) three parameter entities representing volumes;
- (5) stroke text, representing vector-format characters;
- (6) composite entities representing combinations of the above data types.

In the universal data format used in the preferred embodiment of the present invention, the above categories of data are transformed into the corresponding following categories:

- (1) Points are transformed into points.
- (2) Curves are transformed into lines, curves, rational B-spline curves, or linear composites (i.e., multiple overlapping lines).

-21-

- (3) Surfaces are transformed into planes, linear extrusions (i.e., a formula representing a surface generated by a line moving in a prescribed path), or as Non-Uniform Rational B-spline Surfaces (NURBS).
- (4) Solids or volumes are presented as platonic solids, cubic spline surfaces, or in rational B-spline form.
- (5) Stroke text is represented as stroked text, in known fashion.
- (6) Composites are represented as combinations of the above forms.

Most of the transformations described above are straightforward and are known in the art. However, a particularly compact form of data representation is used for the rational B-spline curves, surfaces, and volumes, which are represented as follows:

Eq. 1 Curves with parameter t

$$C(t) = \frac{\sum_{i=0}^n P_i W_i B_{i,m-1}}{\sum_{i=0}^n W_i B_{i,m-1}}$$

Eq. 2 Surfaces with parameters t and s

$$S(t,s) = \frac{\sum_{j=0}^k \sum_{i=0}^n P_{ij} W_{ij} B_{i,m-1} B_{j,l-1}}{\sum_{j=0}^k \sum_{i=0}^n W_{ij} B_{i,m-1} B_{j,l-1}}$$

Eq. 3 Volumes with parameters t, s, u

$$V(t,s,u) = \frac{\sum_{h=0}^r \sum_{j=0}^k \sum_{i=0}^n P_{ijh} W_{ijh} B_{i,m-1} B_{j,l-1} B_{h,p-1}}{\sum_{h=0}^r \sum_{j=0}^k \sum_{i=0}^n W_{ijh} B_{i,m-1} B_{j,l-1} B_{h,p-1}}$$

The P element in each formula represents projective coordinates of control points. The B elements are basis vectors generated using any algorithm based on the common B-spline recurrence relationship (see, e.g., the text reference *A Practical Guide to Splines* by Carl de Boor, Springer 1978). The indices are used as follows: (a) one index i for the control polygon of a curve; (2) two indices i, s for the control net of a surface; and (c) three indices i, s, u for the control volume of a volume. The W element in the expressions is the fourth (homogenous) coordinate of the control points. Thus, the representation of curves, surfaces, and volumes requires a set of 4-dimensional control points, and a set of knot vectors required for the

-22-

generation of the basis functions. These knot vectors are a sequence of floating point values $\{c_0, c_1, c_2, c_3, \dots, c_n\}$ which represent a parameter space upon which the splines are built.

5 The mathematical representation of 3-dimensional information in the universal data format of the present invention permits the usage of conventional algorithms for transformation of the images represented by the model. Thus, by applying known transformation matrices, a graphics image can be panned, zoomed, rotated in any of three dimensions, and otherwise transformed as desired. Examples of conventional transformation matrices are given in U.S. Patent No. 4,862,392, referenced above.

10 The concepts of the present invention can be extended to include the capture of bitmap information into a standard data format. In the prior art, a number of different formats have been used to represent bitmapped information, such as the well known TIFF and BMP formats used in personal computers. Capture of bitmap information extends the capabilities of the present invention to allow the combination of 2-dimensional and 3-dimensional data structures with bitmapped information. This capability is particularly useful in computer systems running the
15 X-Windows protocol, where graphics images are transmitted from a processor to a display device as a bitmap. Conversions from one bitmap format to another bitmap format are well known in the art.

Example of Data Structure Transformation

The table below illustrates a sample transformation of certain entities into the universal data format of the preferred embodiment of the present invention. This table also illustrates the space savings achieved by the present invention by uniquely identifying the representations of certain common entities and comparing them with the common protocol known as PEX.

Geometry entity	PEX Standard Representation	Universal Representation
Line	$\{P_0, P_1\}$	$\{P_0, P_1\}$
Circle or ellipse	$\{P_0, P_1, P_2, P_3, \dots, P_8\},$ $\{t_0, t_1, t_2, t_3, \dots, t_{11}\}$	$\{P_0, P_1, P_2\}, \{FLAG1\}$
Rational cubic B-spline curve	$\{P_0, P_1, P_2, P_3, \dots, P_n\},$ $\{t_0, t_1, t_2, t_3, \dots, t_{n+3}\}$	$\{P_0, P_1, P_2, P_3, \dots, P_n\},$ $\{FLAG2\}$
Polyline with data	$\{P_0, P_1, P_2, P_3, \dots, P_n\},$ $\{c_0, \dots, c_3, \dots, c_n\}$	$\{P_0, P_1, P_2, P_3, \dots, P_n\},$ $\{c_0, \dots, c_3, \dots, c_n\}$
Triangle strip	$\{P_0, P_1, P_2, P_3, \dots, P_n\}$ [i.e., a triangle strip with n-1 vertices]	$\{P_{00}, P_{01}, \dots, P_{10}, \dots, P_{1,n/2-1}\}$
Surface of revolution	$\{P_{00}, P_{01}, \dots, P_{0m}, \dots, P_{n,m-1}, P_{n,m}\},$ $\{t_0, t_1, t_2, t_3, \dots, t_{m+3}\}$ $\{s_0, s_1, s_2, s_3, \dots, s_{n+3}\}$	$\{P'_{00}, P'_{01}, P'_{02}, P'_{11}, \dots, P'_{1,n}\},$ $\{FLAG1\},$ $\{s_0, s_1, s_2, s_3, \dots, s_{n+3}\}$

For all entities except lines, P_n represents a 4 coordinate point x, y, z, w. For lines, w is assumed to be 1, and hence is not stored.

The t_n and s_n values represent knot vectors. The c_n values represent color values. FLAG1 is a 2 byte tag that indicates that the parameters are to be treated by the rendering program as an order 2 Bernstein basis representation. FLAG2 is a 2 byte tag that indicates that the parameters are to be treated as an order 4 B-spline standard representation.

For the triangle strips, the PEX vertex numbering scheme is top to bottom, left to right, whereas the universal format vertex numbering scheme is left to right, top to bottom.

For surfaces of revolution, the points defining the PEX representation are modeled as a set of points that define each surface of revolution, as shown in FIGURE 6. Since a surface of revolution by definition is symmetrical around an axis of revolution 60, any particular plane perpendicular to the axis can be represented by a quarter circle defined by three points (shown as black dots) from which the remaining portion of a rotation circle 61 can be computed. Adjacent rotation circles (e.g., 61-62 or 62-63) are endpoints of a curve that defines the intervening points of the surface (if a rotation circle has a very small or zero radius, the next adjacent rotation circle having a sufficient radius is used).

For most entities, the universal representation achieves substantial space savings. For example, each P_n value (except for lines) is represented as 4 floating point (FP) numbers, and each knot vector and color value is represented as one FP number. A circle or ellipse thus requires 44 ($4 \times 8 + 12$) FP numbers in the PEX representation, but only 12 (3×4) FP numbers plus 2 bytes in the universal representation.

Transmission and Redisplay of Captured Images

Once a capture is complete, a data structure exists that captures substantially all of the relevant geometry information of the original graphics image. This data structure can be accessed by, or transmitted to, another user for redisplay. For example, a first user on a network system can notify a second user that the captured graphics image is available for viewing. If the second user has a compatible viewer function available to his or her computer station 5, 7, that user can simply activate a command or icon to open the data structure into a window on that user's monitor. An interaction component 40 causes an associated application component 44 to access the container holding the relevant data structure within a persistence component 42. The application component 44 then traverses the display list of the captured image in the persistence component 42 and renders the traversed display list as a graphics image compatible with the display characteristics of that user's display device. Thus, the application component 44 takes into account the resolution, color capabilities, and other characteristics of the user's computer station. The second user may then manipulate the displayed image by issuing conventional commands (e.g., rotate, pan, zoom) to the application component 44 through the interaction component 40. Since the underlying geometry information of the original graphics image has

-25-

been preserved in the universal data structure of the captured image, such manipulations can be readily accomplished.

5 The second user can also add additional graphics images to the captured graphics image by using the same process as described above for capturing the first image. The second user can also add annotations to the display captured image by providing additional inputs (e.g., mouse movements, pen movements, voice input, etc.) which are captured in known fashion and stored with the captured graphics image. This annotated image may then be transmitted to the original user or another user. Accordingly, the present invention provides an easy means for obtaining group comment on a drawing without the requirement that each commentator know how to manipulate the originating drawing program or be authorized to run a copy of such a program on their own computer station.

Efficient Rendering

15 Geometry generated during a design process is often the result of the creative thinking of the designer and is not in any particular order or form. For example, the lines and shapes comprising an object can be input in a wide variety of orders. Such shapes and lines are generally represented internally in the order of input. Various colors and styles or other attributes are used as an aid in the creation of an image, and again may be input in almost any order.

20 On the other hand, computer graphics firmware and hardware is generally written to take advantage of "pipelining" effects. Usually, it is most efficient to "prime the pipeline" with the appropriate attributes first, and then render a number of geometric elements having those attributes. For example, if a number of objects have the attribute "red", then those objects are often most efficiently rendered in sequence.

25 Accordingly, the present invention takes advantage of this characteristic by creating bidirectional association between attributes and geometry. This association is created by keeping track of common attributes and re-applying them to new geometry when appropriate. The rendering unit of the present invention uses this additional structure to make efficient use of the underlying hardware. This is done by locating an attribute and priming the rendering pipeline with that

attribute, then running all geometry objects with that attribute through the pipeline. After setting one attribute, multiple geometry nodes or entities having that attribute are processed.

In particular, optimization is preferably accomplished in the following way:

- 5 (1) During conversion from an intermediate data format to the universal format, each object in the traversal tree is examined to determine if it contains attributes. If not, traversal continues.
- (2) If so, then a tree or table of prior processed attribute objects is examined to determine if the attribute in the current object has been used by a prior object.
- 10 (3) If so, then a double-linked pointer is created to link the current object to the prior attribute.
- (4) If not, an attribute object is created to store the attribute information for the current object for future reference.
- (5) Repeat for the next object in the traversal tree.

15 FIGURE 7 shows an un-optimized data structure typical of the prior art. From a root node 70, each geometry object 72 can be accessed by traversing a tree of pointers. Through each geometry object 72, the corresponding attribute object 74 can be reached. As shown, different geometry objects 72 may reference the same type of attribute object 74 (e.g., Attribute 1 or Attribute 2).

20 FIGURE 8 shows an optimized data structure in accordance with the present invention. As should be clear, the data structure in FIGURE 8 is traversed from a new root node 70' through an attribute object 74' first, and then through the geometry objects 72' dependent on that attribute object. The process then repeats for a next attribute object. However, the original root node 70 may also be maintained, so as to allow traversal of the tree through its original linkage relationships.

-27-

Hyper-Connection

Another useful aspect of the present invention is the concept of "hyper-connections". Hyper-connections are used to create "smart" linkages between a combination of captured and created objects. Hyper-connections allow the user to create structured but arbitrary views of a graphics image, thereby enabling communication of graphics images with "rich" content. An example of how to use hyper-connections is shown in FIGURES 9-11.

FIGURE 9 is a pictorial representation of a monitor screen showing an example of usage of the hyper-connection feature of the present invention. The screen display 80 has a drawing portal 82 in which an object 83 is shown depicted. Various transformation controls are provided around the drawing portal 82. The example shown in FIGURE 9 depicts an up/down slider 84, a left/right slider 85, a rotation slider 86, and a zoom slider 87. In addition, other controls may be added, such as for color control, morphing, etc. Transformation controls may also be implemented via menus, text commands, pick lists, etc. Implementation of such controls is well-known in the art, and may be done in software or in hardware. The transformation controls permit different views of a displayed image to be generated.

Also shown is a "channel" button bar 88 that contains at least one active channel selection button 90 as a default button. Buttons added to the channel button bar 88 are "radio" type buttons, in that only one can be active at a time, and selection of one deactivates all others. Included is an connection or note button 89 that activates the hyper-connection procedure.

In FIGURE 10, the graphical image of the object 83 has been zoomed and rotated, as indicated by the movement of the sliders in the rotation slider 86 and the zoom slider 87. To annotate this view, the user would activate the note button 89, which creates a secondary window 93 in which the user may enter a desired annotation. In the illustrated embodiment, the annotation is text, but the annotation may also consist of sound clips, visual clips, graphics, macros, etc., in known fashion. If desired, the user may connect the annotation window 93 to a specific point or area in the image to be annotated, as indicated by the graphic line 95. The annotation 93 and any connection 95, along with the view set on the controls 84, 85, 86, 87, are stored by the hyper-connection system, and associated with a new channel selection button 91, which becomes the

-28-

active button. The previously active channel selection button (in this example, button 90) is visually changed to indicate that it is now inactive.

FIGURE 11 shows the resulting change when the original channel selection button 90 is activated (e.g., by a mouse click), thereby deactivating the previously activated channel selection button 91. All of the controls are returned to their prior positions, the annotation window 93 is removed, and the image of the graphics object 83 is represented exactly as previously seen in that channel.

Of course, multiple annotations of different views can be created, thereby creating additional channel selection buttons in the channel button bar 88. Moreover, the concept of hyper-connections is not limited to annotating only graphical images. Just as the annotations may comprise different types of data objects, the object being annotated may comprise a variety of data objects, such as a bit map, 2D or 3D graphics image, an audio visual clip, an audio clip, program code or functionality, etc.

The hyper-connection feature is preferably implemented as follows:

- (1) Upon loading or capturing an object to be annotated, the state of all controls for the displayed drawing portal 82 is stored and associated with a default channel selection button 90.
- (2) Upon activation of the note button 89 (or similar control, such as a menu selection, hot key, etc.), (a) the state of all controls for the displayed drawing portal 82 is stored and associated with a new channel selection button 91, which becomes active, and (b) an annotation window is displayed.
- (3) User input (e.g., text input, direct input or linkage of audio files, video files, program code, etc.) into the annotation window is accepted and associated with the new channel selection button 91.
- (4) Optionally, user input linking the annotation window to a specific location within the drawing portal is accepted and associated with the new channel selection button 91 (the user may be prompted to create the link prior to acceptance of annotations within the annotation window).
- (5) Activation of another channel selection button causes the stored control state associated with that button to be restored, and the associated annotation and link (if any) to be

-29-

presented (i.e., displayed or, in the case of sound, video, program code, etc., played back).

5 Thus, the preferred embodiment of the hyper-connection feature of the present invention allows a user to capture a graphics image and manipulate it so that it is presented in a plurality of views, any one or more of which may be annotated. Only the view selected, via the channel buttons, is presented at any one time, with any annotation being presented, and, if present, a linkage from the annotation to a portion of the image. That is, the annotations dictate which view of a graphics image is presented to the end user. By selecting different annotations, the end user sees a graphics image transform to the different views defined by the annotating user, each with the corresponding annotation. Accordingly, the invention greatly enhances the ability to communicate information between users without unduly cluttering up the graphics image, such that a sequence of annotated views may be presented to emphasize particular aspects of the image.

15 A number of embodiments of the present invention have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention. Accordingly, it is to be understood that the invention is not to be limited by the specific illustrated embodiment, but only by the scope of the appended claims.

CLAIMS

- 1 1. A method for capturing at least one geometrically-based graphical computer image in a
2 universal format, wherein each geometrically-based graphical image is initially stored in
3 data file format but is rendered by an originating application program into an intermediate
4 data format, comprising:
 - 5 (1) accepting a capture request to capture at least a portion of a geometrically-based
6 graphical image displayed on a visual display device;
 - 7 (2) accessing the intermediate data format for the displayed graphical image;
 - 8 (3) traversing the intermediate data format for the portion of the displayed graphical
9 image to be captured; and
 - 10 (4) translating the traversed portion of the intermediate data format into a universal data
11 format that preserves the geometrical information necessary to render the captured
12 graphical image independently of the originating application program.
- 1 2. The method of claim 1, wherein each graphical image further includes attribute
2 information, and further including the step of translating the traversed portion of the
3 intermediate data format into a universal data format that preserves the attribute
4 information necessary to render the captured graphical image independently of the
5 originating application program.
3. The method of claim 2, further including the step of linking all geometry information to
common attribute information, such that the common attribute information is processed
during rendering before the linked geometry information.

- 1 4. The method of claim 1, further including the steps of:
- 2 (1) capturing rendering-related information generated by an application program upon
- 3 initialization of the application program; and
- 4 (2) using such captured rendering-related information in translating the traversed portion
- 5 of the intermediate data format into a universal data format.
- 6 5. The method of claim 1, wherein the universal data format permits transformations of three-
- 7 dimensional captured graphical images.
- 8 6. The method of claim 1, further including the step of transmitting a translated captured
- 9 graphical image to a remote user for rendering independently of the originating application
- 10 program.
- 11 7. The method of claim 1, further including the step of using a graphical user interface to
- define the portion of a displayed graphical image to be captured and to initiate a capture.
- 1 8. The method of claim 1, wherein the universal data format represents at least some curves
- 2 in the intermediate data format as rational B-spline curves of the form:
- 3

$$C(t) = \frac{\sum_{i=0}^n P_i W_i B_{i,m-1}}{\sum_{i=0}^n W_i B_{i,m-1}}$$

- 4 where P elements represent projective coordinates of control points, the B elements are
- 5 basis vectors generated using any algorithm based on the common B-spline recurrence
- 6 relationship, the W elements are the fourth coordinate of the control points, and the index
- 7 *i* represents a control polygon of the curve.

-32-

9. The method of claim 1, wherein the universal data format represents at least some surfaces in the intermediate data format as non-uniform rational B-spline surfaces of the form:

$$S(t,s) = \frac{\sum_{j=0}^k \sum_{i=0}^n P_{ij} W_{ij} B_{i,m-1} B_{j,j-1}}{\sum_{j=0}^k \sum_{i=0}^n W_{ij} B_{i,m-1} B_{j,j-1}}$$

where P elements represent projective coordinates of control points, the B elements are basis vectors generated using any algorithm based on the common B-spline recurrence relationship, the W elements are the fourth coordinate of the control points, and the indices t, s represent a control net of the surface.

10. The method of claim 1, wherein the universal data format represents at least some volumes in the intermediate data format as rational B-spline volume of the form:

$$V(t,s,u) = \frac{\sum_{h=0}^r \sum_{j=0}^k \sum_{i=0}^n P_{ijh} W_{ijh} B_{i,m-1} B_{j,j-1} B_{h,p-1}}{\sum_{h=0}^r \sum_{j=0}^k \sum_{i=0}^n W_{ijh} B_{i,m-1} B_{j,j-1} B_{h,p-1}}$$

where P elements represent projective coordinates of control points, the B elements are basis vectors generated using any algorithm based on the common B-spline recurrence relationship, the W elements are the fourth coordinate of the control points, and the indices t, s, u represent the control volume of the volume.

-33-

- 1 11. A method for annotating and manipulating information presented on a computer display
2 device, wherein the information presented on the display device can be manipulated to
3 show different views by transformation controls affecting the display device, the state of
4 such transformation controls at a particular time defining a particular view of such
5 information, comprising the steps of:
- 6 (1) storing the state of all transformation controls for an initial view of information on
7 the display device;
- 8 (2) accepting an annotation request to annotate a current view of the information on the
9 display device, where the current view is different from the initial view;
- 10 (3) accepting at least one annotation to be associated with the current view of the
11 information on the display device;
- 12 (4) storing the state of all transformation controls for the current view of the information
13 on the display device, and each associated accepted annotation;
- 14 (5) providing a selection control for each stored state; and
- 15 (6) upon selection of a selection control corresponding to a stored state, then:
- 16 (a) restoring the transformation controls to the state stored in such stored state,
17 such that the view of the information defined by such stored state is presented
18 on the display device; and
- 19 (b) presenting any annotation associated with such stored state.

-34-

- 1 12. A method for annotating and manipulating a graphics image displayed on a computer
2 display device, wherein an image displayed on the display device can be manipulated to
3 show different views by transformation controls affecting the display device, the state of
4 such transformation controls at a particular time defining a particular view of such image,
5 comprising the steps of:
- 6 (1) storing the state of all transformation controls for an initial view of a graphics image
7 on the display device;
- 8 (2) accepting an annotation request to annotate a current view of the graphics image on
9 the display device, where the current view is different from the initial view;
- 10 (3) accepting at least one annotation to be associated with the current view of the
11 graphics image on the display device;
- 12 (4) storing the state of all transformation controls for the current view of the graphics
13 image on the display device, and each associated accepted annotation;
- 14 (5) providing a selection control for each stored state; and
- 15 (6) upon selection of a selection control corresponding to a stored state, then:
- 16 (a) restoring the transformation controls to the state stored in such stored state,
17 such that the view of the graphics image defined by such stored state is
18 presented on the display device; and
- 19 (b) presenting any annotation associated with such stored state.

1 / 11

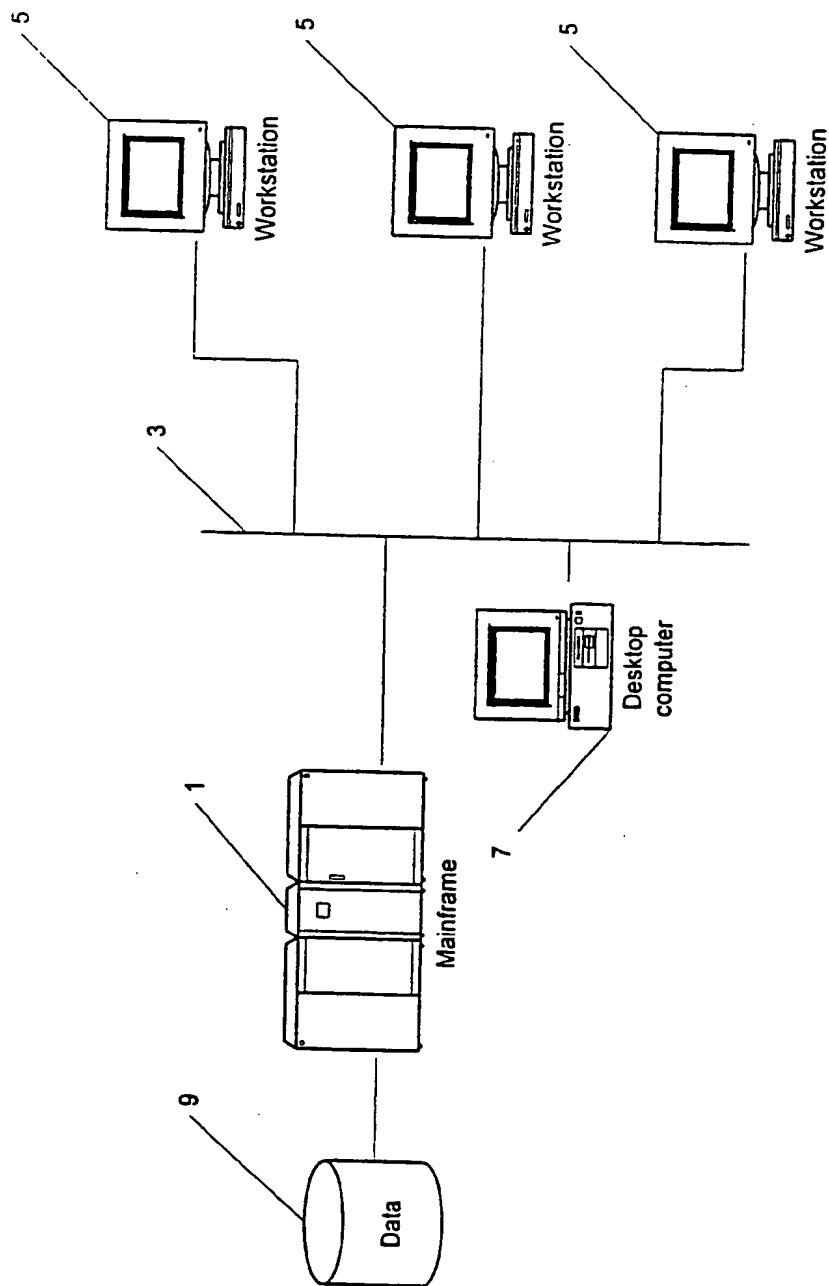


FIG. 1
(Prior Art)

2 / 11

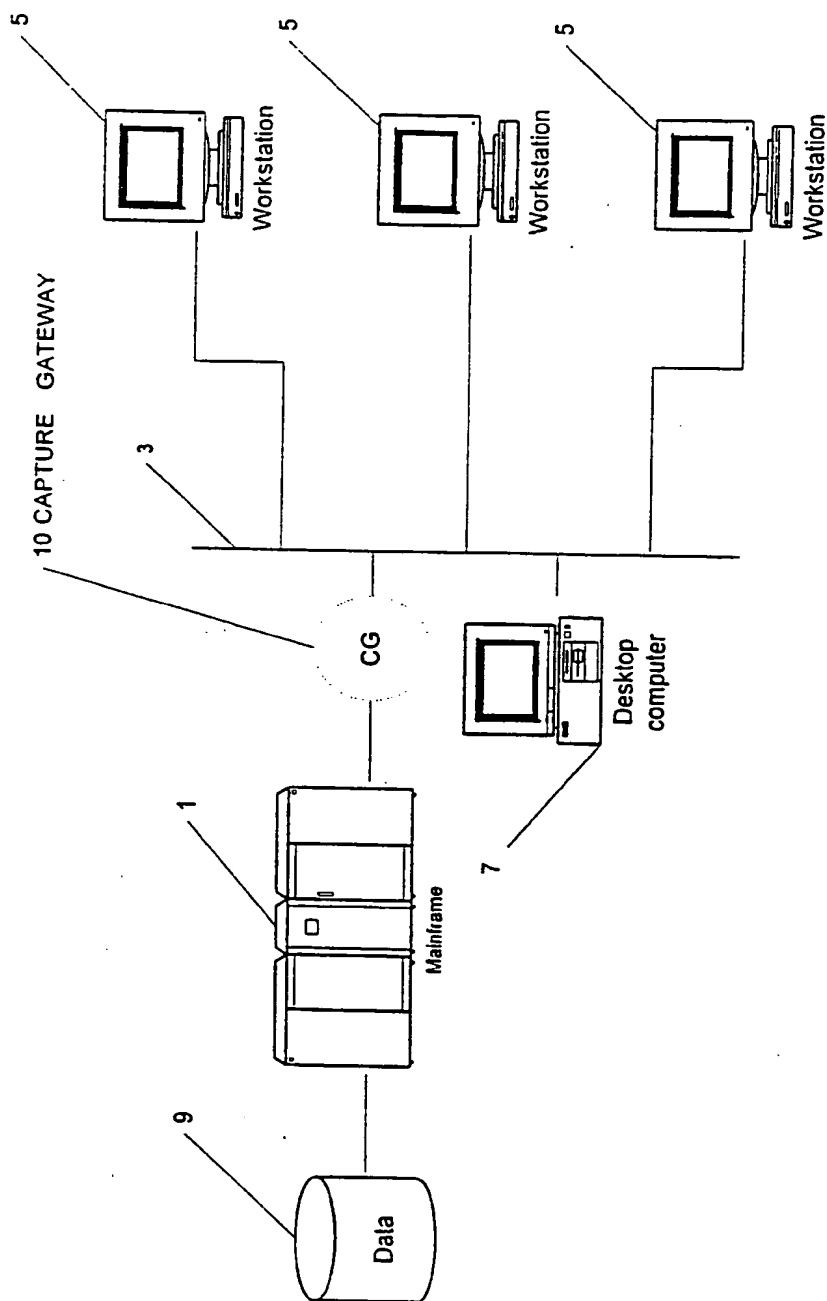


FIG. 2

3 / 11

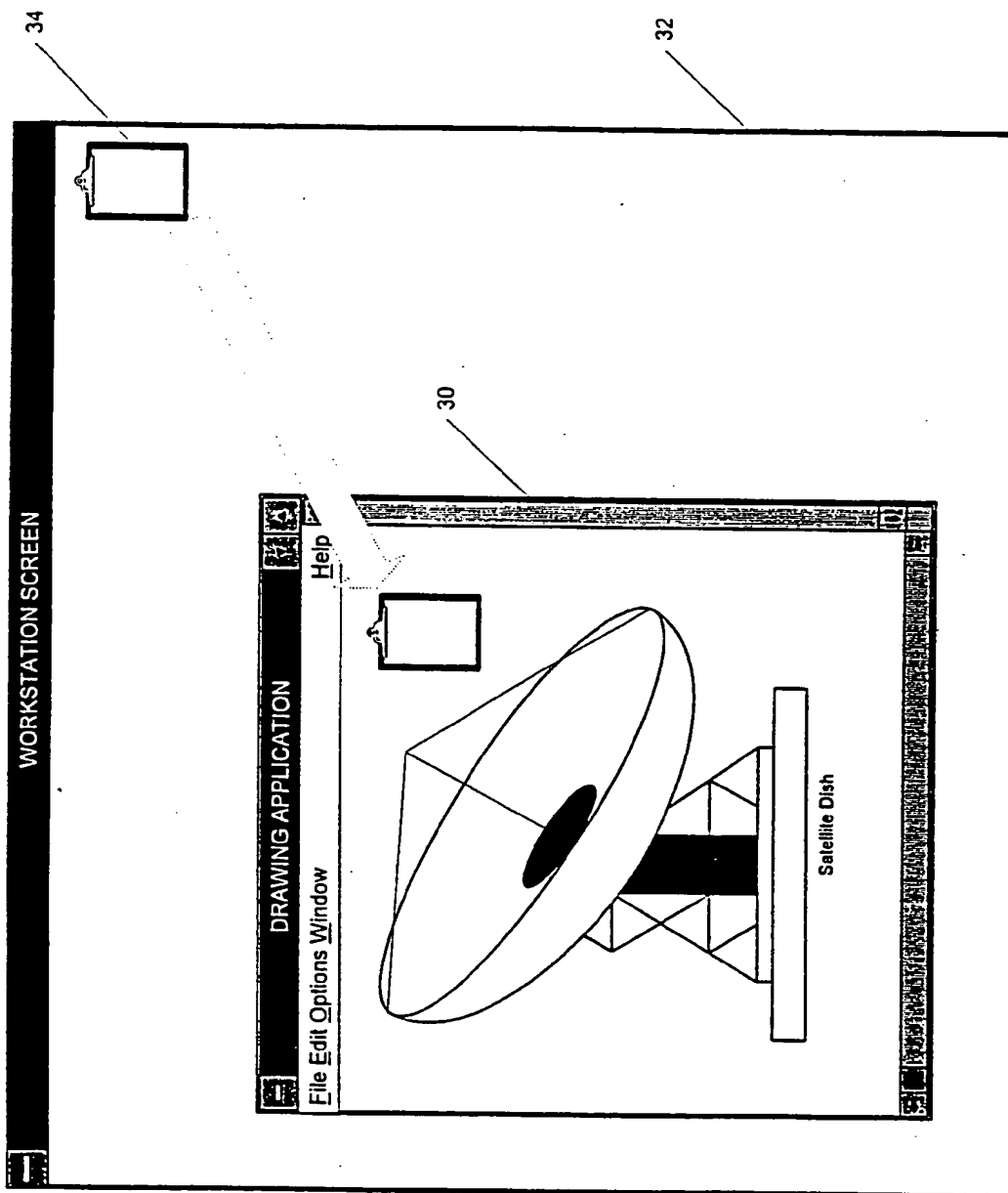


FIG. 3

4 / 11

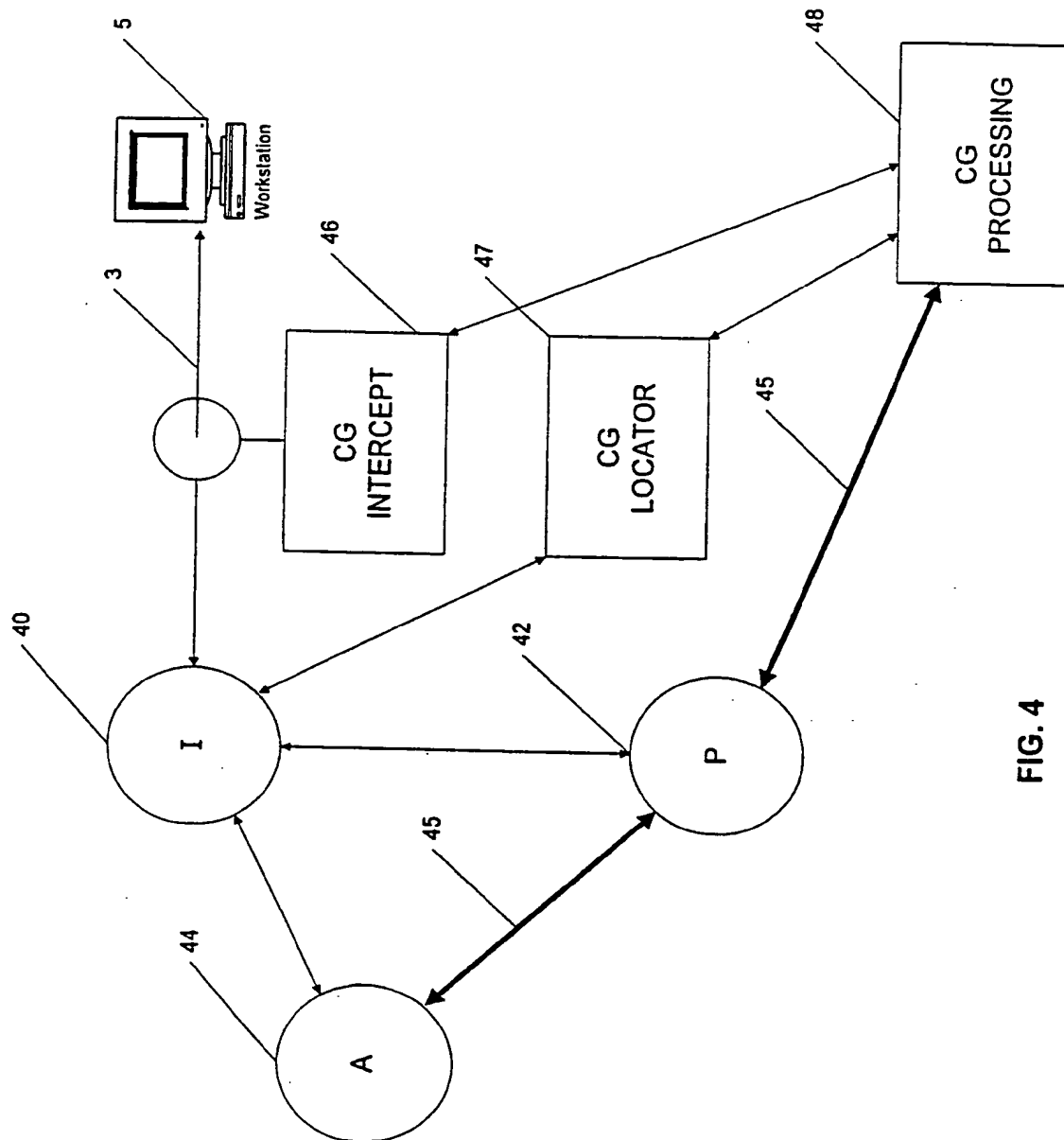


FIG. 4

5 / 11

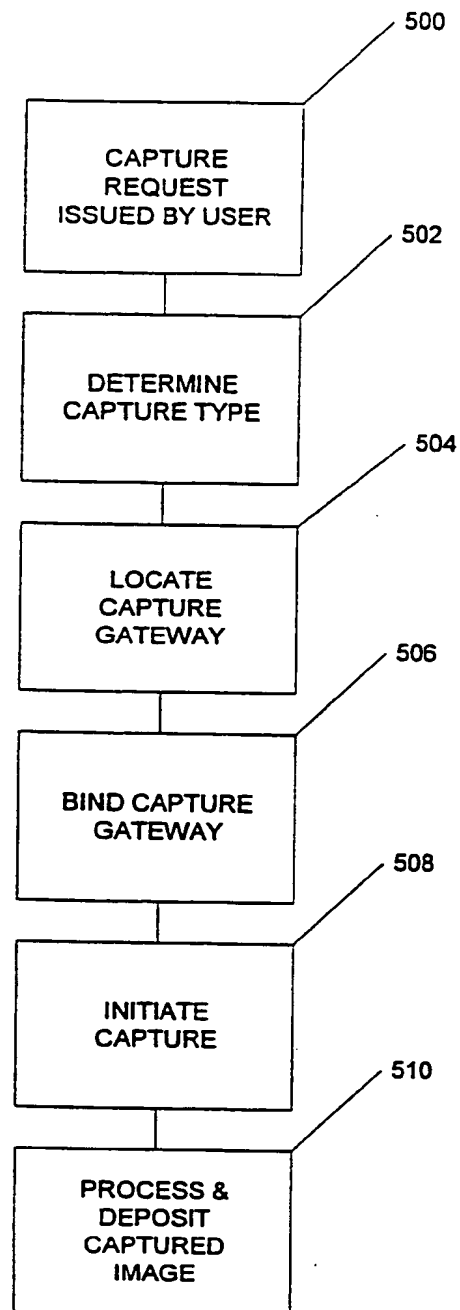


FIG. 5

6 / 11

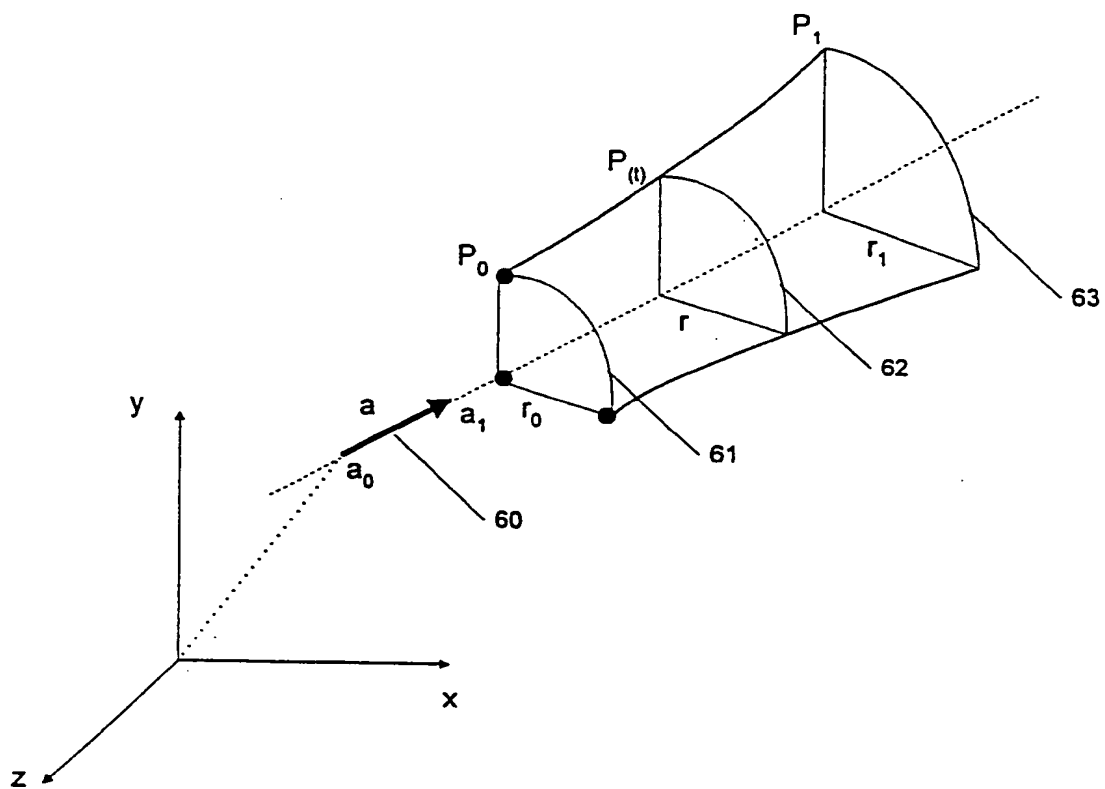


FIG. 6

7 / 11

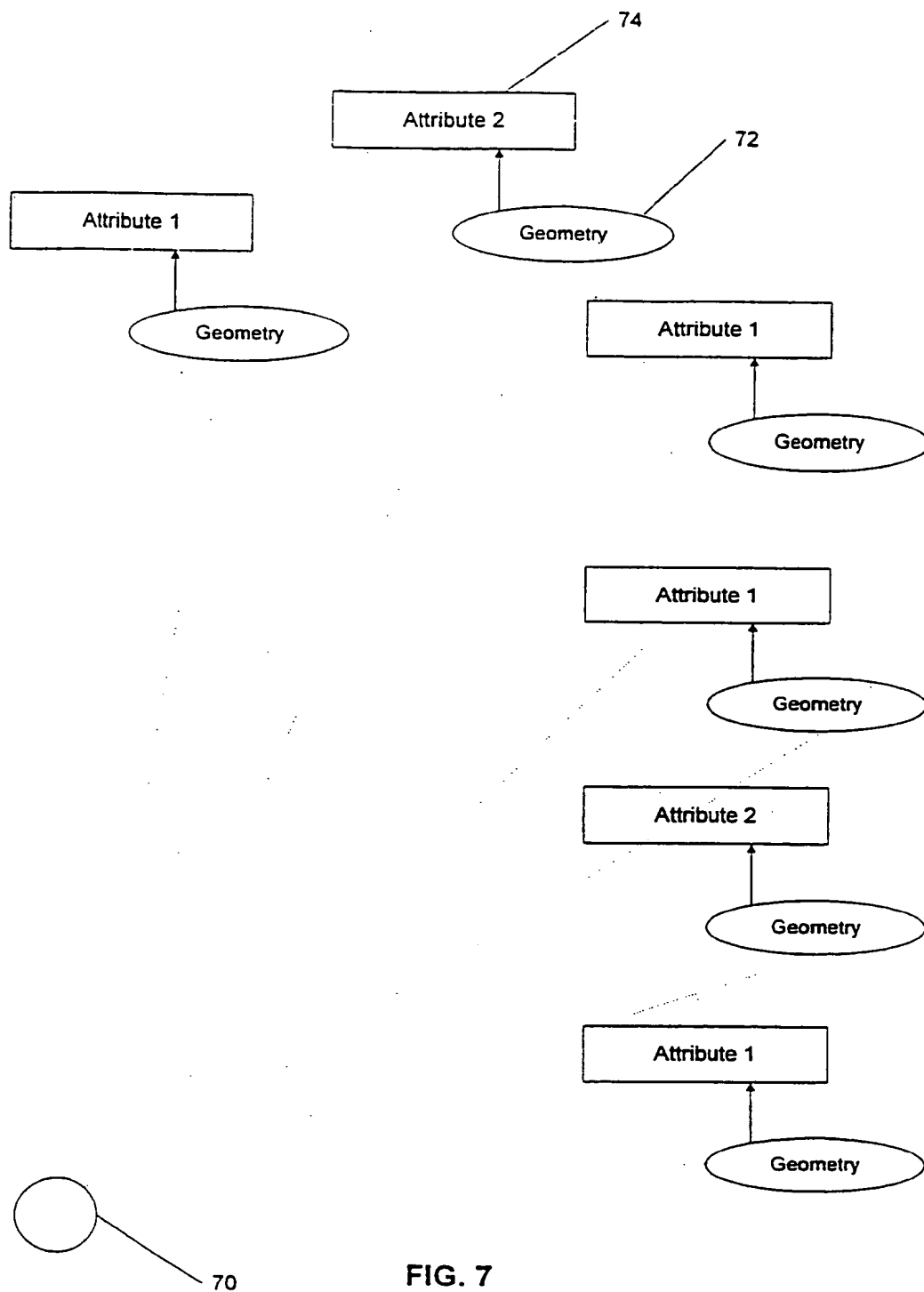


FIG. 7

8 / 11

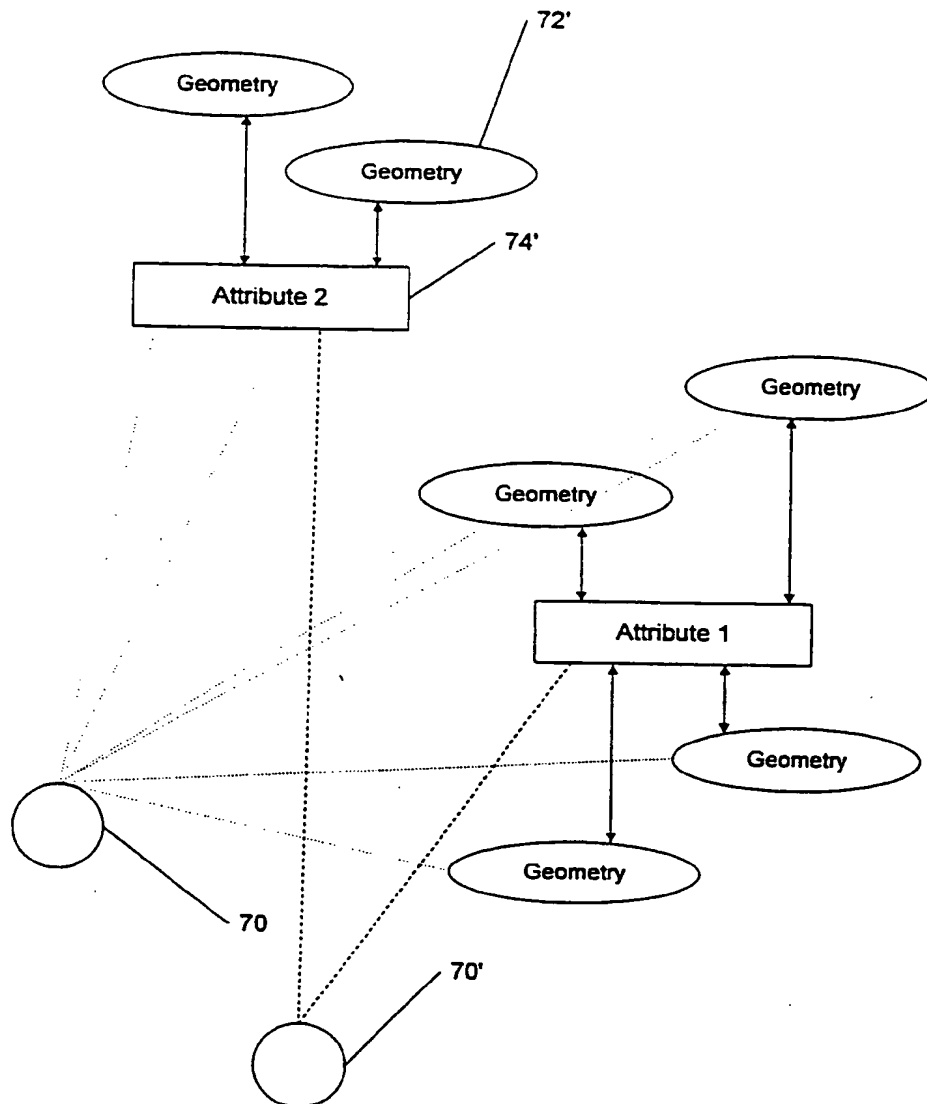


FIG. 8

9 / 11

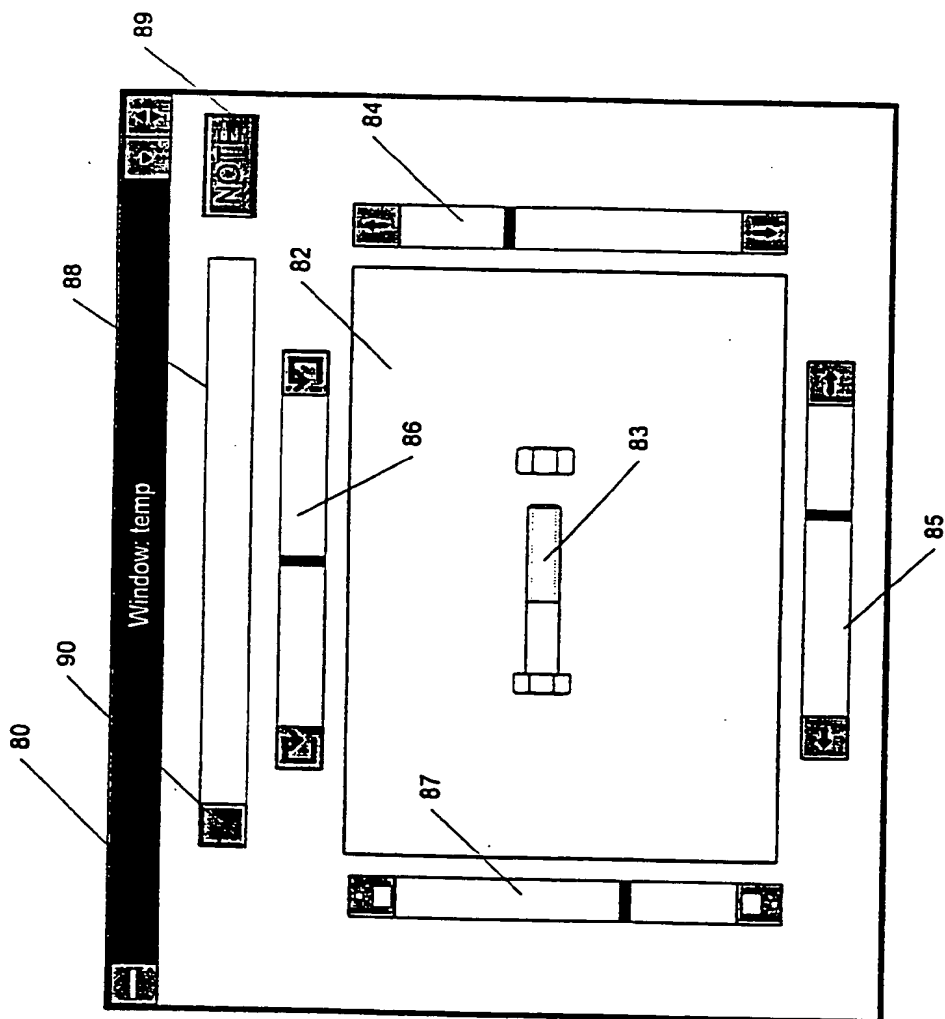


FIG. 9

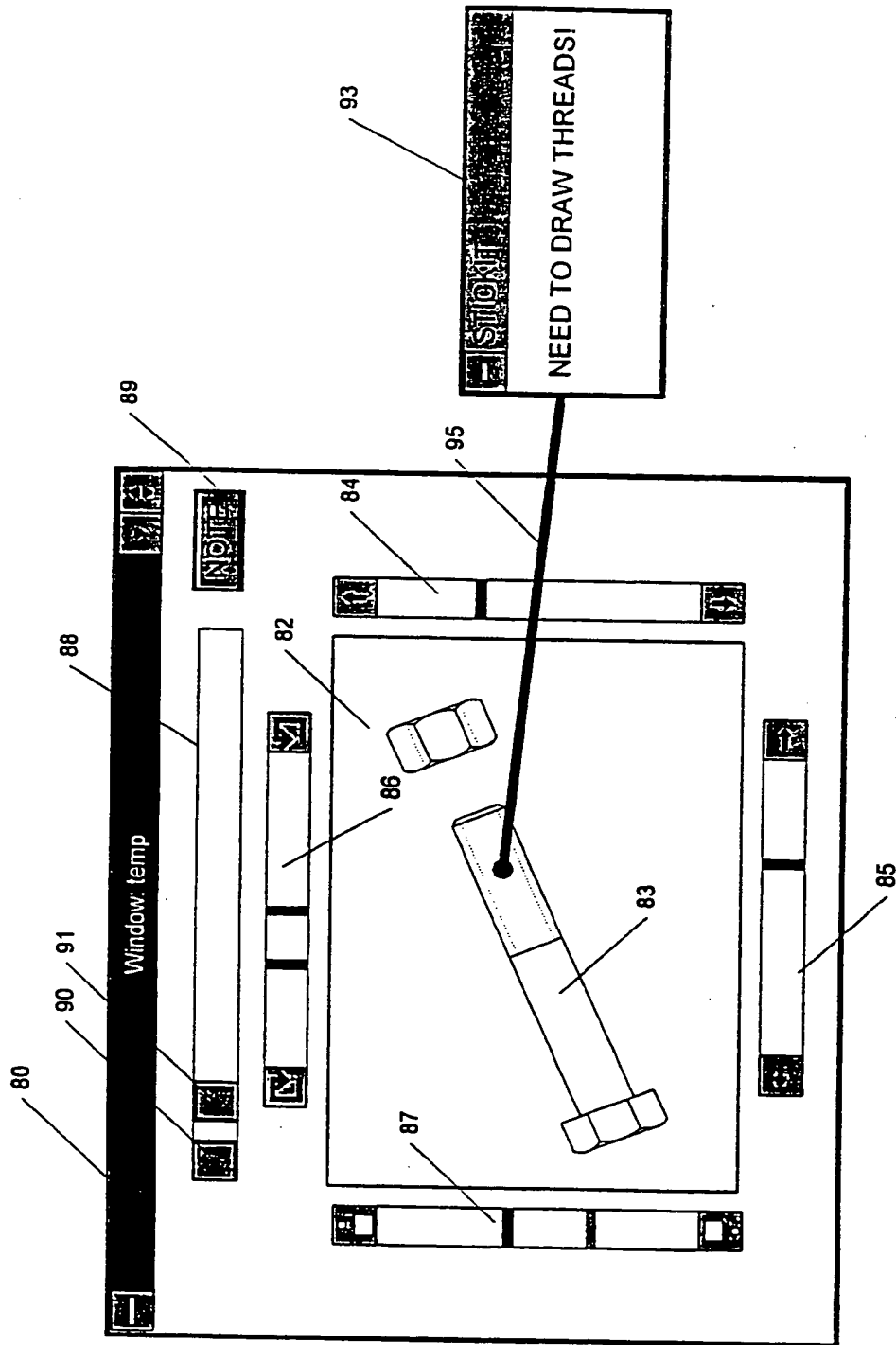


FIG. 10

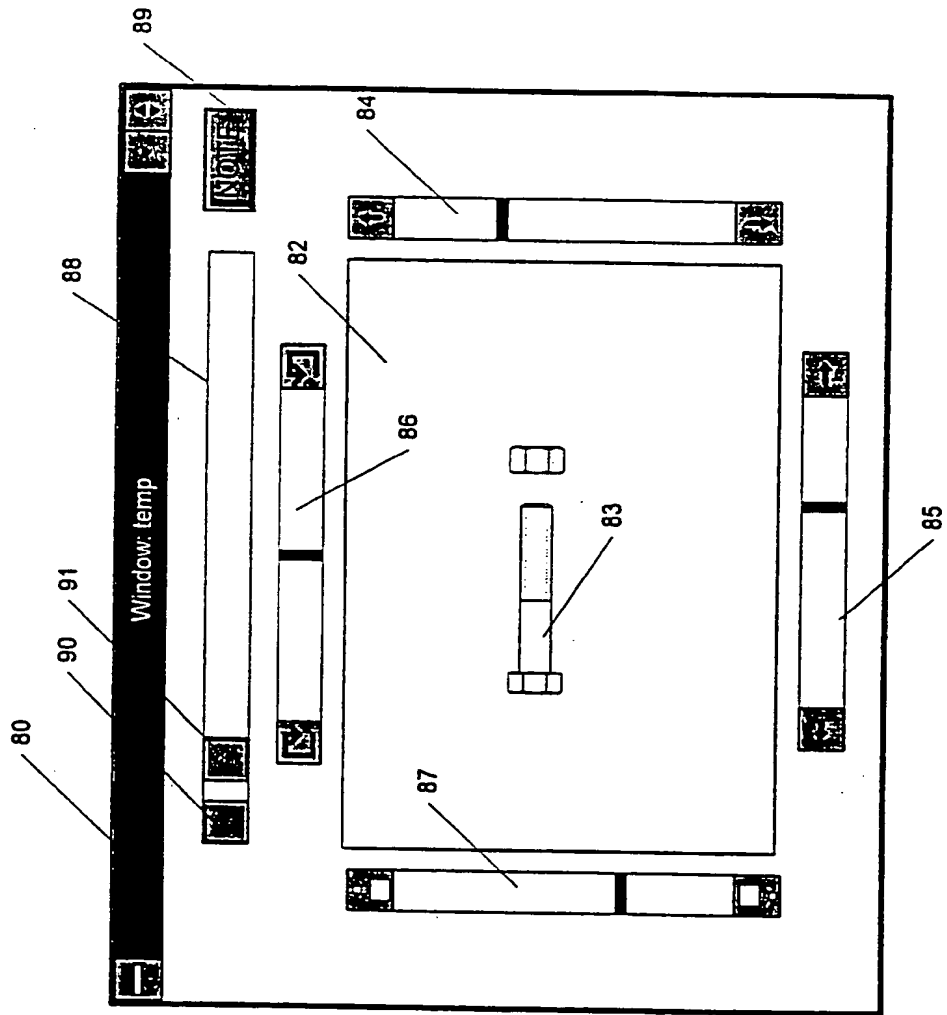


FIG. 11

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US95/07210

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) : G06T 9/00, G06F 17/30

US CL : 395/142; 364/419.19

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 395/142, 161; 364/419.19, 419.02, 226.4, 920.4; 382/305

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Foley et al., Computer Graphics: Principles and Practice, 1990

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS, IEEE/IEE Publications on Disc

search terms : data conversion, translation, annotation, help, state, b-spline

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US, A, 5,083,262 (HAFF, JR.) 21 January 1992 Fig. 1 & 2, Abstract	1-7
X,P	US, A, 5,339,434 (RUSIS) 16 August 1994 Fig. 1, Abstract	1-7
X	US, A, 4,897,799 (LE GALL ET AL.) 30 January 1990 Fig 1, Abstract	1-7
X,P	US, A, 5,392,393 (DEERING) 21 February 1995 Fig. 3, Abstract	1-7
Y	US, A, 5,317,682 (LUKEN, JR. ET AL.) 31 May 1994 Abstract, Col. 4-5 equations	8-10
A	US, A, 5,241,654 (KAI ET AL.) 31 August 1993 Col. 3 equations	8-10

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:	*T	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
A document defining the general state of the art which is not considered to be part of particular relevance	*X*	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
E earlier document published on or after the international filing date	*Y*	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	*Z*	document member of the same patent family
O document referring to an oral disclosure, use, exhibition or other means		
P document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search

03 AUGUST 1995

Date of mailing of the international search report

06 OCT 1995

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

ANTON FETTING

Telephone No. (703) 305-9646

Form PCT/ISA/210 (second sheet)(July 1992)*

INTERNATIONAL SEARCH REPORT

 International application No.
 PCT/US95/07210

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US, A, 5,239,373 (TANG ET AL.) 24 August 1993 col. 6 lines 9-28	1-7
X	US, A, 5,138,697 (YAMAMOTO ET AL.) 11 August 1992 Abstract, Figs. 4, 5, 6, col. 2 line 30 to col. 3 line 68	1-7
X,P ----- Y,P	US, A, 5,390,292 (TAKAMURA ET AL.) 14 February 1995 Abstract, Fig. 1, col. 3 lines 45-58, col. 4 lines 33-45, Col. 4-5 equ. 1 Col. 4-5 equ. 1	1-9 ----- 10
Y,P	US, A, 5,404,295 (KATZ ET AL.) 04 April 1995 Abstract, col. 3 line 60 - col. 4 line 8	11-12
Y	US, A, 5,103,498 (LANIER ET AL.) 07 April 1992 Abstract, Fig. 3A,	11-12
Y	US, A, 5,175,812 (KRIEGER) 29 December 1992 Abstract, Fig. 2 & 3	11-12
Y	US, A, 5,231,578 (LEVIN ET AL.) 27 July 1993 Abstract, Fig. 3 & 8, Col. 2 col. 20-45	11-12
Y	US, A, 5,309,359 (KATZ ET AL.) 03 May 1994 Abstract, Col. 2 lines 54-66	11-12
Y,P	US, A, 5,371,675 (GREIF ET AL.) 06 December 1994 col. 3 lines 50-59	11-12
X	M. Sakauchi, "Two Interfaces in Image Database Systems", 10-12 April 1989, International Workshop on Industrial Applications of Machine Intelligence & Vision, p. 22-27, especially p. 24	1-7
Y	Thingvold et al., "Physical Modeling with B-Spline Surfaces for Interactive Design and Animation" 25-28 March 1990, Proceedings 1990 Symposium on Interactive 3D Graphics, p. 129-137, especially p. 130	8-10
Y	Liu et al., "CAD-Based Automated Machinable Feature Extraction", Signals, Systems & Computers, 1993 27th Asilomar Conference, 1993, p. 558-562, especially p. 559	1-7
A	M. Hill, "File Format for Data Exchange Between Graphic Data Bases", DALDAC78 Conference, 1978, p. 54-59	1-7

Form PCT/ISA/210 (continuation of second sheet)(July 1992)*

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US95/07210

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	Spooner et al., "Engineering Data Exchange in the ROSE System", Intelligent Control 1990 International Symposium, 1990, p. 972-976, especially p. 973	1-7
Y	Zobrist, "Investigation of IGES for CAD/CAE Data Transfer, Circuits and Systems", 1990 IEEE Midwest Symposium, 1990, p. 902-905, especially p. 902	1-7
A	Ho et al., "IGES and PDES, The Current Status of Product Data Exchange Standard", Computer Integrated Manufacturing, 1988 International Conference, 1988, p. 210-216	1-7
Y	Futrelle, "The Conversion of Diagrams to Knowledge Bases", Visual Languages, 1992 ,IEEE Workshop, 1992, p. 240-242	1-7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US95/07210

BOX II. OBSERVATIONS WHERE UNITY OF INVENTION WAS LACKING

This ISA found multiple inventions as follows:

This application contains the following inventions or groups of inventions which are not so linked as to form a single inventive concept under PCT Rule 13.1. In order for all inventions to be examined, the appropriate additional examination fees must be paid.

Group I, claims 1 to 10, drawn to methods of converting graphical data into a universal format.

Group II, claims 11 and 12, drawn to methods of attaching annotations to process states and saving the annotations with the states for subsequent retrieval.